NAMA : EDWARD **Mata Kuliah : Research Topics In Computer Science**
NIM : 2201741971 **Kode Mata Kuliah : COMP6575**
KELAS : LB-08 **Fakultas / Departemen : School of Computer Science**

I. ESAI (40%)

1. **(20%) Mandiri** - Jelaskan yang disebut dengan self-plagiarism! Menurut Anda, apa saja faktor yang dapat menyebabkan munculnya self plagiarism? Berikan contoh paper dalam area ilmu komputer yang memiliki indikasi self-plagiarism, serta berikan penjelasan indikasi self-plagiarism pada contoh Anda.

Jawaban:

Self plagiarism

- ➔ Adalah **menyalin makalah penelitian Anda sendiri yang telah Anda lakukan di masa lalu**. Ini **bisa berupa beberapa baris atau gagasan yang pernah Anda gunakan sebelumnya**.
- ➔ Dalam self plagiarism, **hal-hal seperti penerbitan dua makalah identik juga termasuk yang disebut duplikat publikasi**.
- ➔ **Meskipun kita menyalin karya kita sendiri**, itu **tidak menolak fakta bahwa kita harus belajar atau memahami sesuatu yang baru dengan setiap makalah penelitian**.

Faktor-faktor yang dapat menyebabkan plagiarisme diri dalam makalah penelitian:

a) Ketika saya menerbitkan makalah penelitian, makalah itu milik jurnal yang saya terbitkan. Jadi jika saya menggunakan beberapa kata atau kata-kata sendiri tanpa izin dari jurnal kita dapat menyebabkan plagiarisme diri untuk diri kita sendiri.
b) Mengambil ide dari makalah penelitian sebelumnya dan membangun makalah penelitian baru berdasarkan itu.

Area yang memiliki indikasi plagiarisme sendiri dalam ilmu komputer adalah "Algoritma Perbandingan". **Karena banyak makalah penelitian** tentang ilmu komputer **tentang mengurangi kompleksitas suatu algoritma**.

Contohnya suatu Algoritma Perbandingan pada awalnya memiliki kompleksitas O(n^2), lalu pada perubahan dan perkembangan selanjutnya, kompleksitas menjadi O(log(n), maka kompleksitas menjadi semakin kecil dan proses menjadi semakin cepat.

2. **(20%) Mandiri** - Jelaskan apa yang dimaksud dengan novelty pada sebuah jurnal! Berikan contoh novelty dari 2 paper referensi yang digunakan dalam penelitian Anda (pastikan paper yang digunakan sebagai contoh berbeda pada setiap anggota kelompok) serta berikan penjelasan mengapa bisa Anda katakan bahwa paper tersebut memiliki unsur novelty. Note: lampirkan paper yang digunakan sebagai contoh ketika mengumpulkan jawaban Anda.

Jawaban:

Novelty

➔ **Merupakan suatu kebaharuan dari suatu hal yang pernah dibuat, terjadi, atau tersedia**.Kemajuan pesat dalam dua dekade terakhir karena munculnya dan aksesibilitas teknologi baru yang memungkinkan berbagi barang dan data. Akibatnya, **mungkin sulit untuk menemukan topik yang tidak diketahui atau tidak tersedia literatur**. **Namun masih ada banyak ruang untuk kemajuan**.

➔ **Juga bisa terletak pada kontradiksi dengan sesuatu yang dilaporkan sebelumnya**. Survei literatur menyeluruh termasuk analisis makalah penelitian dan paten perlu dilakukan pada topik yang menarik **untuk memastikan bahwa ada beberapa hal baru yang terkait dengannya**. **Variasi dalam metodologi yang dilaporkan sebelumnya yang menghasilkan hasil yang bervariasi juga dapat dianggap sebagai NOVELTY**.

2 Paper Referensi yang memiliki unsur Novelty:

a) [8] Nasrollahi, Kamal, Sergio Escalera, Pejman Rasti, Gholamreza Anbarjafari, Xavier Baro, Hugo Jair Escalante, and Thomas B. Moeslund. (2015) "Deep learning based super-resolution for improved action recognition." In Image Processing Theory, Tools and Applications (IPTA), 2015 International Conference on, pp. 67- 72. IEEE. DOI: 10.1109/IPTA.2015.7367098

LINK: https://scihub.wikicn.top/https://ieeexplore.ieee.org/abstract/document/7367098

Judul Paper: Deep Learning based Super-Resolution for Improved Action Recognition

File PDF Paper ini sudah saya berikan di ZIP-Nya, KLIK sesuai Judul Paper-Nya

NAMA : EDWARD Mata Kuliah : Research Topics In Computer Science
NIM : 2201741971 Kode Mata Kuliah : COMP6575
KELAS : LB-08 Fakultas / Departemen : School of Computer Science

Alasan Mengapa Mengandung Unsur Novelty:

1) Menggunakan Algoritma Resolusi yang sudah ada

To produce upscaled high-resolution videos, that are less affected by artifacts, from low-resolution recordings, super-resolution algorithms have been used. These algorithms can be applied in both spatial and frequency domains [1]. The frequency domain approaches are more suitable for cases where motions of the objects follow simple models like translation. Examples of such cases are satellite imaging. For surveillance videos in real-world scenarios spatial domain approaches are more suitable as they can cope with the complicated motion structures of objects in the scene. These algorithms are generally divided into two groups: single-image-based and multiple-image-based super-resolution algorithms [1], [2].

2) Menjelaskan Tantangan dari Algoritma yang sudah ada diterapkan di topic mereka

Multiple-image super-resolution algorithms, like [3], receive a couple of low-resolution images of the same scene as input and usually employee a registration algorithm to find the transformation between them. This transformation information is then used along with the estimated blurring parameters of the input low-resolution images, to combine them into a higher scale framework to produce a super-resolved output image. For multiple-image super-resolution algorithms to work properly, there should be sub-pixels displacements between input low-resolution images. Furthermore, these sub-pixels displacements should be estimated properly by the registration algorithm, which is usually a challenging task, especially when complicated motion of non-rigid objects, like human body, needs to be modeled. These algorithms are guaranteed to produce proper higher resolution details, however, their improvement factors are usually limited by factors close to two [1].

3) Menjelaskan Kelebihan dan Kelemahan Algoritma yang digunakan

Single-image super-resolution algorithms, like [4], do not have the possibility of utilizing sub-pixel displacements, because they only have a single input. Instead, they employee a kind of training step to learn the relationship between a set of high-resolution images and their low-resolution counterparts. This learned relationship is then used to predict the missing high-resolution details of the input low-resolution images. Depending on the relationship between the training low-and high-resolution images, these algorithms can produce high-

NAMA : EDWARD **Mata Kuliah : Research Topics In Computer Science**
NIM : 2201741971 **Kode Mata Kuliah : COMP6575**
KELAS : LB-08 **Fakultas / Departemen : School of Computer Science**

4) Algoritma banyak digunakan di berbagai ruang ligkup aplikasi

Super-resolution algorithms have been applied to a wide range of applications, such as face recognition [5], [6], biometric recognition [7], and target recognition [8]. However, they have not been used for activity recognition in low-resolution videos. Such an application of super-resolution seems very important, because due to the distance between cameras and subjects of interest, in real-world scenarios, subjects occupy only a small portions of each frame, which makes activity recognition very challenging. Fig. 1 shows the importance and need for an upsampling algorithm in low-resolution images for action recognition algorithms. It is shown in this figure that, regardless of the targeted action, the recognition accuracy of action recognition algorithms drop considerably as the resolution of the images drop.

5) Menggambarkan perkembangan sistem dari algoritma yang sudah ada

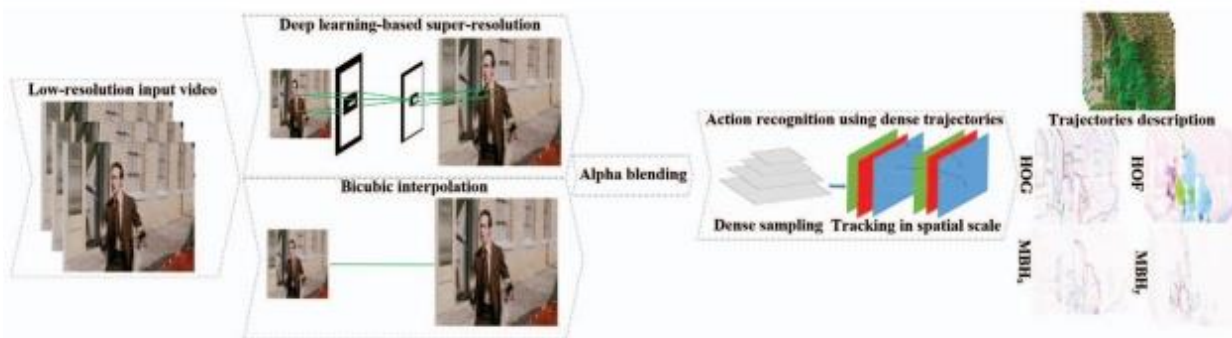
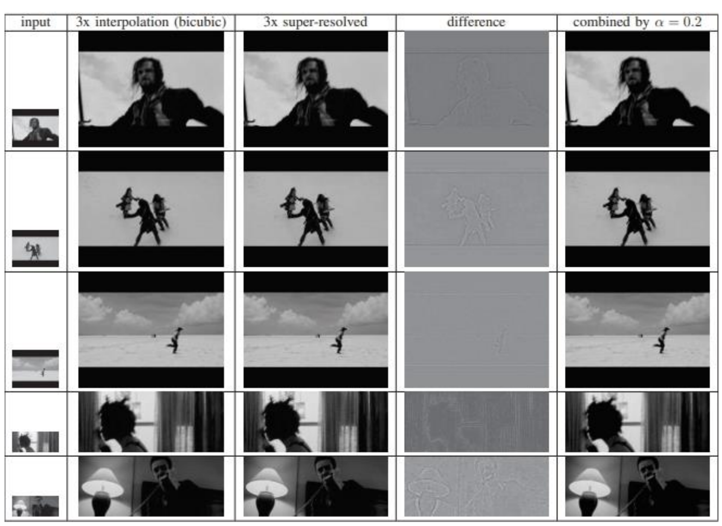


Fig. 2: The block diagram of the proposed system.

6) Memperlihatkan Bukti Perbedaan dengan Hasil yang mudah dipahami



b) [9] Savioja, Lauri, Akio Ando, Ramani Duraiswami, Emanuel AP Habets, and Sascha Spors. (2015) "Introduction to the issue on spatial audio." IEEE Journal of Selected Topics in Signal Processing 9, no. 5: 767-769. DOI: 10.1109/JSTSP.2015.2447112

LINK: https://ieeexplore.ieee.org/document/7154533

Judul Paper: Introduction to the Issue on Spatial Audio

File PDF Paper ini sudah saya berikan di ZIP-nya, KLIK sesuai Judul Paper-nya

Alasan Mengapa Mengandung Unsur Novelty:

1) Topik ini sudah popular beberapa tahun lalu

SPATIAL audio is an area that has gained in popularity in the recent years. Audio reproduction setups have evolved from the traditional two-channel stereophonic setup towards multi-channel loudspeaker setups. Advances in acoustic signal processing even made it possible to create surround sound listening experiences as well as attempts to create binaural real life listening experiences using traditional stereo speakers and headphones. Finally, there has been an increased interest in creating different sound zones in the same acoustic space. At the same time, the computational capacity provided by mobile audio playback devices has increased significantly. These developments enable new possibilities for advanced audio signal processing, such that in the future we can record, transmit and reproduce spatial audio in ways that have not been possible before. In addition, there have been fundamental advances in our understanding of 3D audio.

2) Sudah banyak Paper menjelaskan Standard Audio

duction systems for spatial audio, ranging from headphones to 22.2 speakers, it is a major challenge to ensure interoperability between formats and systems, and consistent delivery of high quality spatial audio. Therefore, the MPEG committee has established a new standard for 3D audio coding. The first paper by Herre *et al.* in this special issue is an invited paper that introduces the MPEG-H 3D Audio standard and the underlying principles.

NAMA : EDWARD **Mata Kuliah : Research Topics In Computer Science**
NIM : 2201741971 **Kode Mata Kuliah : COMP6575**
KELAS : LB-08 **Fakultas / Departemen : School of Computer Science**

3) Sudah ada Paper Sebelumnya mengenai topik ini

tion. In previous works it has been shown that prior knowledge about the room can be exploited by dictionary-based localization methods. Chardon *et al.* show that these methods fail to accurately localize a source when the measurements are done at frequencies close to modal frequencies of the room. The authors propose a new model for the acoustic sound field in case both the room geometry and boundary conditions are unknown using the Vekua theory that allows to decompose the sound field in a direct and reverberant sound field. Based on this model a dereverberation pre-processing step is developed that allows to remove the reverberation from acoustical measurements without any prior on the room or on the signals. This pre-processing step is compatible with various localization methods, and enables the narrowband source localization in a unknown reverberant environment.

4) Menggunakan referensi paper mengenai Wireless Communication

The paper of Dokmanic *et al.* assumes that some methods for locating early reflections for the source of interest are available, and then seeks to develop various beamforming algorithms that can take advantage of this knowledge. Using an inexact analogy with ideas of rake receivers in wireless communication, they term their beamformers as an "acoustic rake receivers". They then extend several standard beamforming frameworks such as delay-and-sum, minimum variance distortionless response, and Signal-to-Interference-and-Noise Ratio in various ways to incorporate information about the reflections. While

5) Menjelaskan ada teknik tradisional, dan solusi modern untuk perkembangan zaman

Besides the traditional techniques for spatial sound reproduction, like stereophony or Sound Field Synthesis, parametric techniques aim at the dynamic decomposition of a captured sound field into spatio-temporal components. These components are often inspired by human spatial perception. A well known representative of these methods is Directional Audio Coding. Politis *et al.* present an extension of the method which provides a higher separation between simultaneous sources and reverberation. It is based on a sectorial analysis of the sound field using spherical harmonics signals which may have been captured by an spherical microphone array. The results of a

LINK VIDEO PRESENTASI: https://youtu.be/rZwOXU359yI

Image Analysis With R Programming

LINK VIDEO PRESENTASI:
https://youtu.be/rZwOXU359yI

Aldo Jonathan Handaka
Computer Science
Binus University
Jakarta, Indonesia
aldo.handaka@binus.ac.id

Evan Edbert
Computer Science
Binus University
Jakarta, Indonesia
evan.edbert@binus.ac.id

Aeron Mitchell
Computer Science
Binus University
Jakarta, Indonesia
aeron.mitchell@binus.ac.id

Edward
Computer Science
Binus University
Jakarta, Indonesia
edward007@binus.ac.id

Abstract— The R programming language excels at facilitated assignments. It is broadly utilized as a high level, free and open source language which is re-mark ably unique, deciphered, scripting and multi-world view. It additionally underpins object situated programming highlights and can be utilized as a general purpose programming language. R is simpler to learn and has more straight forward linguistic structure when contrasted with C, C++ and Java. R is similarly popular for work area based applications. The fields where R truly sparkles in are information science and machine learning, numeric, representative computations. Also, it is utilized in different fields like Image processing, Games, Web improvements and Big Data Analytics. Picture preparing with R is an exceptionally efficient and viable procedure for doing tasks such as dissecting the digitization of the pictures to extract the required data. A few activities such as improving the quality, upgrading, zooming, blurring, inverting the picture, composing content on the images, greyscale, performing picture rebuilding, recovering, etc. is conceivable with R. Here's the challenge; given two images, determine if one image is a subset of the other image. This will be valuable for solving this present reality undertakings and procedures in a very effective way.

Keywords—image, analysis, subset, R Programming, processing

I. Introduction

In our daily life, people use picture as a tool for many things such as memory recovery, new discovery of people and places, certain disease study, crime investigation, map making, and else. Due to how important this picture as a tool in our life, this concept has become a necessity for developers to create a program that can differentiate a lot of images to a certain degree to help human life. As human technology becomes more progressive, there will be one time when an algorithm will be created to differentiate image from one another despite their complexity and structure. This fore-mentioned algorithm is what people called as image analysis and it becomes the fundamental basis of recognizing and differentiating images. Nowadays, image analysis is used for many of daily aspects such as Comparison of image quality with different retargeting methods quickly and reliably [4], Estimation of solid depth map of a single monocular natural image [5], Presentation of a series of psychophysical experiments to determine the simultaneous dynamic series of human visual systems under the full adaptation of background lighting [6], Development of mentoring system that notifies the driver if the speed is insufficient according to the visibility conditions [7], Image-based Illumination Inspired by Using Decomposition of Local Singular Value and Discrete Wavelet Transformation [8], and image processing applied in digital sound system [9]. For example, a face recognition application is installed within our phones to recognize our faces and store them in the storage. [2]

A. *Why to learn and use R?*

R is an elevated level programming language. R translators are accessible for some working frameworks that permit the execution of R code on a wide assortment of frameworks. R bolsters various programming standards, which incorporates object arranged, practical programming and procedural style. It oversees memory consequently and has an enormous and complete. In 1993 Ross Ihaka and Robert Gentleman created the R language. R is gotten from numerous other programming dialects, for example, C, C++, Unix shell and so on. [1]

B. *Why Image Analysis?*

An image is merely a visual portrait of something. This means that it can represent a person, an animal, or any living or non-living thing. A picture is basically a rectangular grid of pixels with a definite height and width. Each pixel possesses its own value. Thus, image quality depends on the values of this pixel, and pixel is the information unit present in an image. Image Processing is image enhancement utilizing mathematical operations for which the input is an image, such as an image or video clip and the image processing result can be either an image or a collection of image-related characteristics or parameters.

Need of image processing :

- Humans are not satisfied with the quality of images and therefore they make use of image processing.
- Humans rely upon their visual system (eyes and brain) to collect visual information about their surroundings. Visual information refers to images and videos. In the past, we needed visual information mainly for survival. Nowadays, visual information is required for survival as well as for communication and entertainment purpose
- To enhance an image
- To extract some useful information from an image that can be utilized for heath sciences, public safety, etc.

[3] So, in short, following steps are involved in image processing:

a. Input image
b. Segmented map before integration
c. Edge map before integration
d. Segmented map and edge map after Integration
e. Pixel clustering

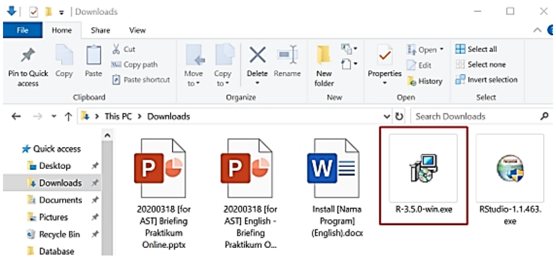
C. Why to make use of R for Image Analysis?

R has multiple packages for multiple purposes like web development, scientific and numeric computing, image analysis. To work on images, R has a packages i.e. R Imaging Library (RIL) for image analysis operations. The R Imaging Library provides many functions for image analysis. We performed some basic operations using RIL modules.

II. Methodology

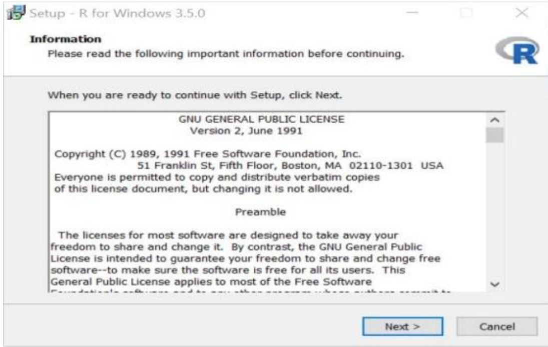
A. Installing R Language 3.5.0 on Windows

a. Click on R-3.5.0-win.exe to start the installer

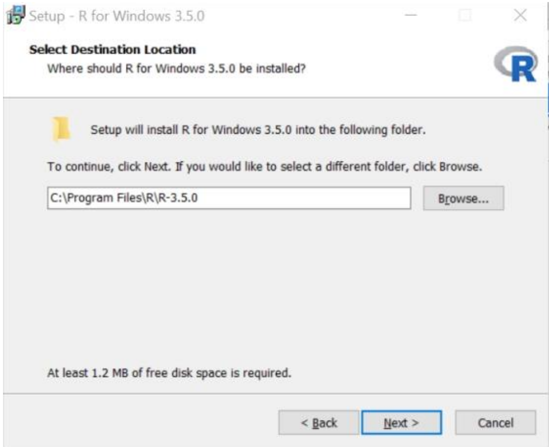


b. Select your preferred language (default english) then click OK

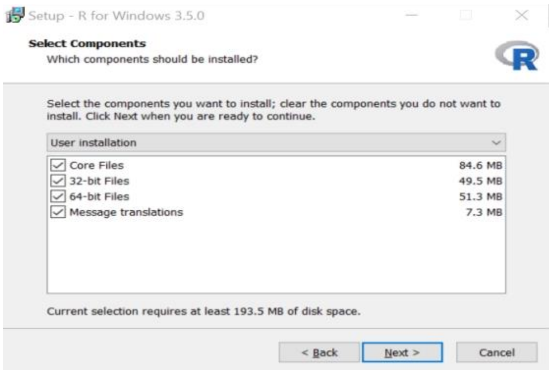
c. Click Next on terms & agreement



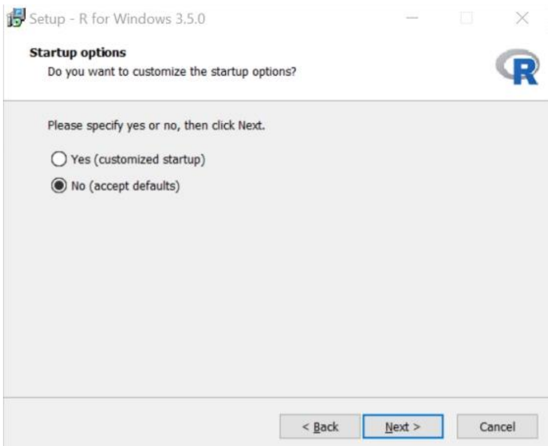
d. Select your preferred installation location then click Next



e. Click Next on select file to install

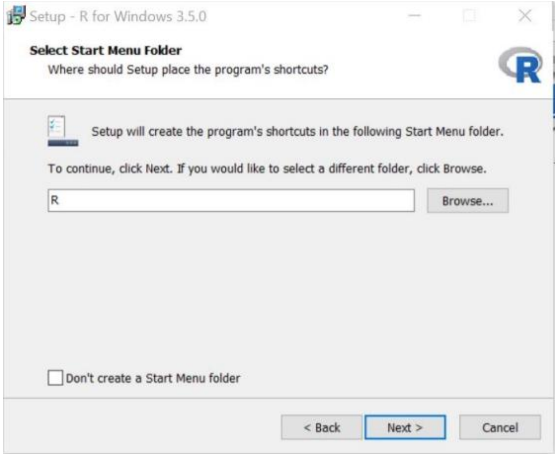


f. Select No to accept the default configuration then click Next

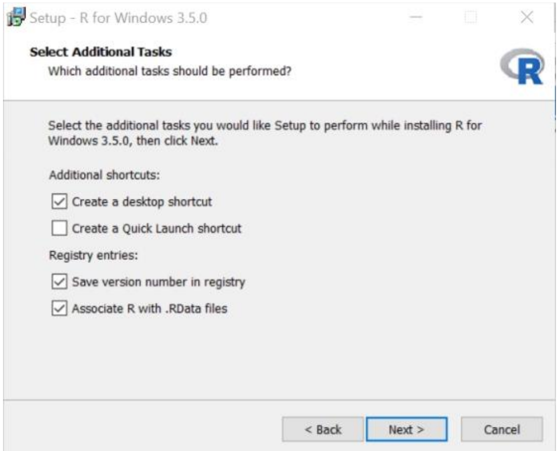


LINK VIDEO PRESENTASI: https://youtu.be/rZwOXU359yI

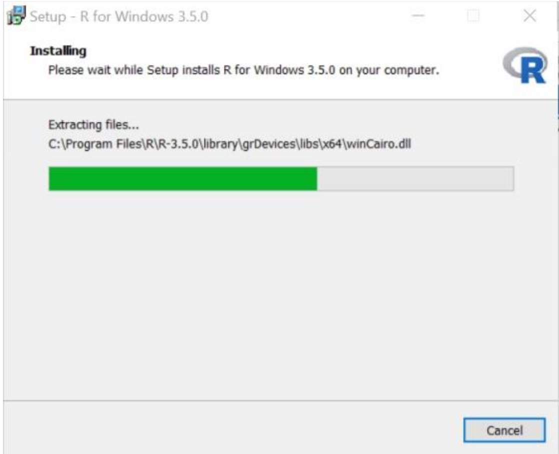
g. Click Next



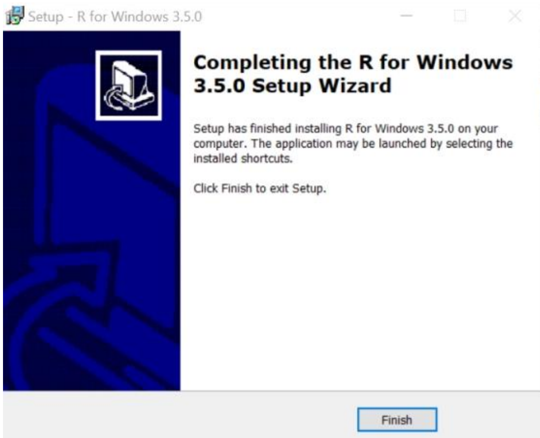
h. Select additional task that you want then click Next



i. Wait the process until completed

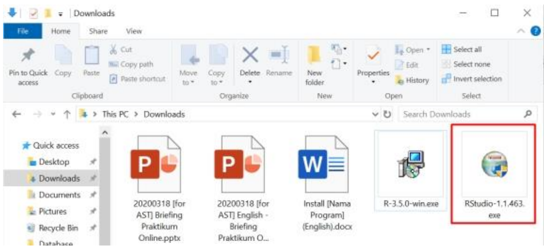


j. Click Finish



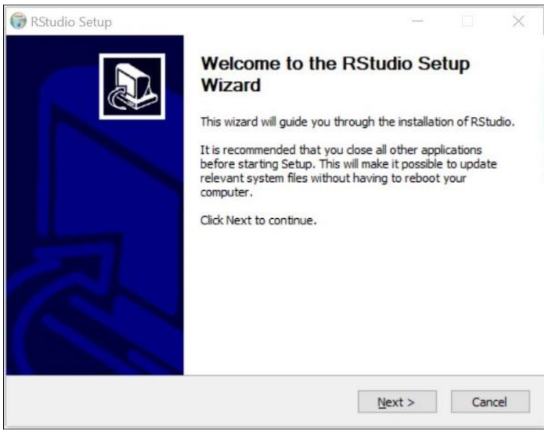
B. Installing Rstudio 1.1.463 on Windows

a. Click on RStudio-1.1.463.exe to start the installer

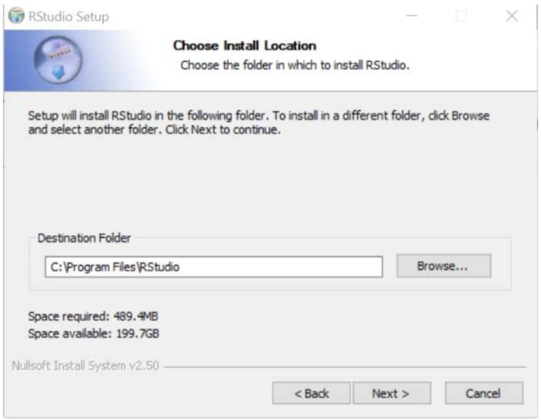


LINK VIDEO PRESENTASI: https://youtu.be/rZwOXU359yI

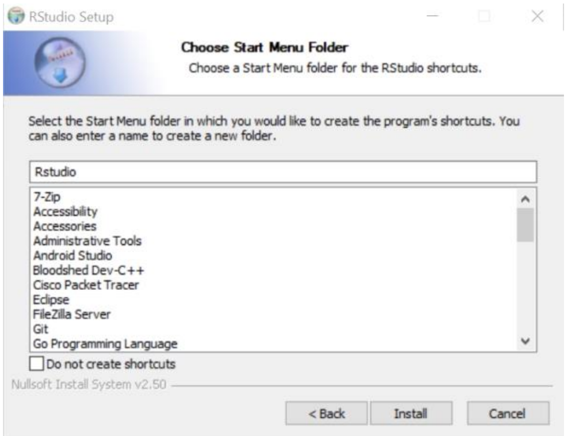
b. Click Next



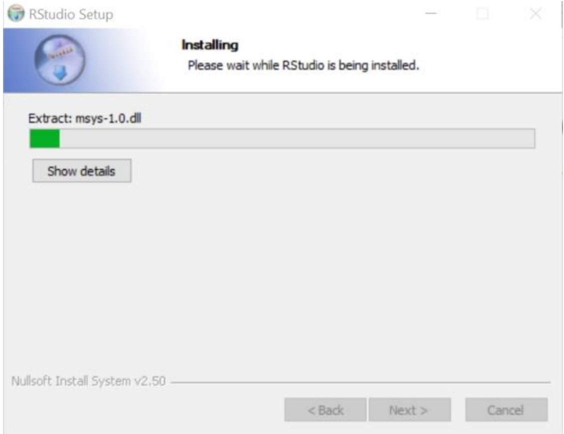
c. Select your preferred installation location then click OK



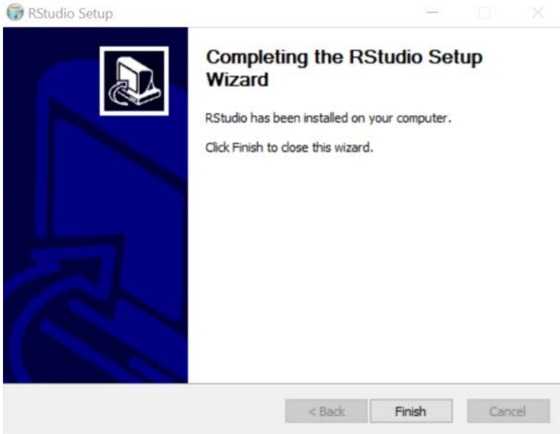
d. Click Next



e. Wait for the process to complete



f. Click Finish to complete the installation



C. Code Implementation

1) DataSet that will be used

- 103992931.jpg
- 168680522.jpg
- 460249177.jpg
- 460249177a.jpg
- 477899129.jpg
- 478946583.jpg
- 478946583a.jpg
- 78771293.jpg
- 78771293a.jpg

2) Install Package and Library (Zoom to 350%)

```
# http://cran.r-project.org/web/packages/ripa/index.html
install.packages("ripa")
library(ripa)
# doesn't appear to provide access to a subset of a matrix. Documentation unclear how to use jpeg

# # # # # #
# EBImage
-------
source("http://bioconductor.org/biocLite.R")
biocLite("EBImage")
install.packages("tiff") #package ‘tiff’ is not available (for R version 3.1.2)
library("EBImage")
# Error in loadNamespace(j <- i[[1L]], c(lib.loc, .libPaths()), versionCheck = vI[[j]]) :
# there is no package called ‘tiff’
# Error: package or namespace load failed for ‘EBImage’
# ubuntu solution at http://mytechscribblings.wordpress.com/2013/06/28/installing-ebimage-package-for-r-using-rstudio-in-ubuntu/
browseVignettes("EBImage")
f = system.file("images", "sample.png", package="EBImage")
img = readImage(f)

# # # # # #
# grep, grepRaw, grepl
install.packages("jpeg")
library(jpeg)
img.one <- readJPEG("2_CodeClinic_imageAnalysis/imagesToAnalyze/460249177.jpg")
img.one.subset <- readJPEG("2_CodeClinic_imageAnalysis/imagesToAnalyze/460249177a.jpg")
grepl(img.one.subset[,,1], img.one[,,1], useBytes=TRUE)
grepRaw(as.vector(img.one.subset[,,1]), as.vector(img.one[,,1]), value=FALSE)

# Warning message:
# In grepl(img.one.subset[, , 1], img.one[, , 1], useBytes = TRUE) :
#  argument 'pattern' has length > 1 and only the first element will be used

# # # # # #
# package Raster
#http://cran.r-project.org/web/packages/raster/index.html
install.packages("raster")
library(raster)

# # # # # #
# package rimage
install.packages("rimage")
# Warning in install.packages : package ‘rimage’ is not available (for R version 3.1.2)
```

LINK VIDEO PRESENTASI: https://youtu.be/rZwOXU359yI

3) *Plotting The Rasters (Zoom to 350% to clear view)*

```
# testing plotting functions

plotThisJpeg <- myjpeg
ptj.width <- ncol(plotThisJpeg)
ptj.height <- nrow(plotThisJpeg)
plotThisClip <- myjpegclip
ptjc.width <- ncol(plotThisClip)
ptjc.height <- nrow(plotThisClip)

plot(c(0, ptj.width), c(0, ptj.height), type = "n")
# plot(plotThisJpeg,useRaster=TRUE)
# plot(plotThisClip,add=TRUE,useRaster=TRUE)
rasterImage(plotThisJpeg, 0, 0, ptj.width, ptj.height)

rasterImage(plotThisJpeg[1:1500,1:2500,1], 0, 0, ptj.width, ptj.height)

rasterImage(plotThisClip, 500, 500, ptjc.width, ptjc.height)

this <- arrayInd(points.of.interest[1000],dim(haystack.raster))
xleft <- this[2]
ybottom <- this[1]
rect(xleft,ybottom,(ptjc.width+xleft),(ptjc.height+ybottom))

corltHayToNeedle <- function (aPOI) {
  needle.height.offset <- needle.quarter.height + aPOI
  needle.width.offset <- needle.quarter.width + aPOI
  ccf(haystack.raster[aPOI:needle.height.offset,1:needle.width.offset,1],needle.brick)
}

applyResults <- lapply(points.of.interest,corltHayToNeedle)
```

4) *Calculate Cross Correlation (Zoom to 350 %)*

```
# demonstration of cross correlation

# set up dataX - we'll keep reusing this vector
dataX <- 1:10

# cross correlation for different values of dataY
dataY <- dataX # 100% cross correlation
dataY <- rnorm(10)*10 # not 100% correlation
dataY <- rev(dataX) # reverse the values in dataX

# show dataX and dataY side-by-side
data.frame(dataX,dataY)

# show a plot
plot(dataX,type="l",col="red")
lines(dataY,col="blue")

# calculate the cross-correlation
ccf(dataX,dataY)

# plot showing lag of 2 between dataX and dataY

dataYLag2 <- c(dataY[3:10],dataY[1:2]) # first, build a lag of dataY
data.frame(dataX,dataY,dataYLag2)      # show the values
plot(dataX,type="l",col="red")         # plot the values
lines(dataYLag2,col="blue")
ccf(dataX,dataYLag2)                   # cross-correlate
```

```
  stepSize <- 100
  for (rowIndex in seq(from=poi.rows[1],
                       to=poi.rows[length(poi.rows)],
                       by=stepSize)) {
    for (columnIndex in seq(from=poi.columns[1],
                            to=poi.columns[length(poi.columns)],
                            by=stepSize)) {
      #      if (rowIndex+needle.height < haystack.height &
      #          columnIndex+needle.width < haystack.width) {
      haystack.subset.to.be.correlated <- haystack.red[
        (rowIndex:(rowIndex+needle.height)),
        (columnIndex:(columnIndex+needle.width))
        ]

      ccf.object <- ccf(as.vector(haystack.subset.to.be.correlated),
                        as.vector(needle.red),
                        plot=FALSE)
      max.ccf <- max(ccf.object$acf)
      ccf.max <<- append(ccf.max,max.ccf)
      xpos <<- append(xpos,columnIndex)
      ypos <<- append(ypos,rowIndex)

      ##########
      # creates pretty graphics - and slows things down
      maintitle <- paste("ccf=",round(max.ccf,2),"xpos=",columnIndex,"ypos=",rowIndex)
      maintitle2 <- paste("ccf=",round(max.ccf,2))
      saveHere <- paste0("Plot",sprintf("%04d",rowIndex),sprintf("%04d",columnIndex),"_x",columnIndex,"_y",rowIndex,".

      png(filename=saveHere)
     # nf <- layout(matrix(c(1,2),1,2,byrow=T),c(1,1),c(1,1),T)
      #layout.show(nf)
      #boxplot(ccf.object$acf,ylim=c(0,1),main=maintitle2)
      plot(c(0, haystack.width), c(0, haystack.height), main=maintitle,type = "n")

      #        rasterImage(haystack.subset.to.be.correlated,
      #                    xleft=0,
      #                    xright=ncol(haystack.subset.to.be.correlated),
      #                    ytop=nrow(haystack.subset.to.be.correlated),
      #                    ybottom=0)
      rasterImage(haystack.raster,
                  xleft=0,
                  xright=haystack.width,
                  ytop=haystack.height,
                  ybottom=0)

      rasterImage(needle.red,
                  xleft=columnIndex,
                  xright=(needle.width+columnIndex),
                  ytop=(haystack.height-(rowIndex+needle.height)),
                  ybottom=(haystack.height-rowIndex))
      dev.off()
      ##########

      cat("rowIndex=",rowIndex,"columnIndex=",columnIndex,"           \r")
      #      }
    }
  }

  ccf.results <- matrix(c(xpos,ypos,ccf.max),ncol=3)
  colnames(ccf.results) <- c("xpos","ypos","ccf.max")

}
```

5) *Function to check if a Image is subset of another Image (Zoom to 350% to clear view)*

```
# function to test if one image is a cropped version of another
# needle - image that might be a cropped version
# haystack - image that might be a parent version
# returns TRUE or FALSE

isthisacroppedversionofthat <- function(needle,haystack) {

  #   needle <- paste("imagesToAnalyze/","460249177a.jpg",sep="")
  #   haystack <- paste("imagesToAnalyze/","460249177.jpg",sep="")

  # assumes needle and haystack are jpeg images
  needle.raster <- readJPEG(needle)
  needle.width <- ncol(needle.raster) # width and height are used a lot
  needle.height <- nrow(needle.raster)
  haystack.raster <- readJPEG(haystack)
  haystack.width <- ncol(haystack.raster)
  haystack.height <- nrow(haystack.raster)
  red.layer <- 1 # We're only interested in one layer
```

```
  # if needle is larger than or same size as haystack,
  # then needle can't be a cropped version.
  if ((needle.height >= haystack.height)
      && (needle.width >= haystack.width)) return(c(0,0))

  # I'm going to assume that a match in one (of RGB) layers between
  # needle and haystack is a match between all layers. This reduces the
  # amount of data we need to correlate by 2/3
  # This requires versions of needle and haystack with just one layer
  needle.red <- needle.raster[,,red.layer]
  haystack.red <- haystack.raster[,,red.layer]

  # points.of.interest is an array of x,y coordinates that are possible
  # starting points for a subset image. Any points outside of this range
  # are either too narrow or too shallow to fit needle.
  # remember that we are talking about rows and columns of the IMAGE...
  # ...NOT the graph. Images start with 0,0 at upper left.
  # Graphs start with 0,0 in lower left
  diff.haystack.needle.height <- haystack.height-needle.height
  poi.rows <- (1:diff.haystack.needle.height)

  diff.haystack.needle.width <- haystack.width-needle.width
  poi.columns <- (1:diff.haystack.needle.width)

  xpos <<- numeric()
  ypos <<- numeric()
  ccf.max <<- numeric()
```

6) *Testing Cod e if Worked or Not Worked (Zoom to 350%)*

```
# set up for test

install.packages("jpeg")
library(jpeg)

setwd("2_CodeClinic_ImageAnalysis")
source("isthisacroppedversionofthat.R")

funcWrapper <- function(anImagePair) {
  needle <- paste("imagesToAnalyze/",anImagePair[1],sep="")
  haystack <- paste("imagesToAnalyze/",anImagePair[2],sep="")
  is.it.a.match <- isthisacroppedversionofthat(needle,haystack)
  if (is.it.a.match[1] > 0) {
    print(paste(anImagePair[1],"is a subset of",anImagePair[2],"with correlation of",is.it.a.match[1],"and at point",is.it.a.match[2]))
  } else {
    print(paste(anImagePair[1],"is not a subset of",anImagePair[2]))
  }
}

# test the function
list.of.images <- list.files("imagesToAnalyze")
image.pairs <- expand.grid(list.of.images,list.of.images)
apply(image.pairs,1,funcWrapper)


# test known subsets
funcWrapper(c("460249177a.jpg","460249177.jpg")) #this is a subset
funcWrapper(c("478946583a.jpg","478946583.jpg")) #this is a subset
funcWrapper(c("78771293a.jpg","78771293.jpg")) #this is a subset
funcWrapper(c("103992931.jpg","78771293.jpg")) #this is not a subset
funcWrapper(c("168680522.jpg","103992931.jpg")) #this is not a subset
```

7) Main Code (zoom to 350%)

```
install.packages("jpeg")
library(jpeg)

setwd("2_CodeClinic_ImageAnalysis")

needle.raster <- readJPEG("imagesToAnalyze/460249177a.jpg")
needle.width <- ncol(needle.raster) # width and height are used a lot
needle.height <- nrow(needle.raster)
needle.brick <- needle.raster[,,1]

haystack.raster <- readJPEG("imagesToAnalyze/460249177.jpg")
haystack.width <- ncol(haystack.raster)
haystack.height <- nrow(haystack.raster)

# discovering the row/column location where these two images match
# places needle on top of haystack
needle.x <- 1450
needle.y <- 1690

plot(c(0, haystack.width), c(0, haystack.height),
     type = "n",main=paste("x=",needle.x,"y=",needle.y))

rasterImage(haystack.raster,
            xleft=0, xright=haystack.width,
            ytop=haystack.height,ybottom=0)

rasterImage(needle.brick,
            xleft=needle.x,
            xright=needle.x+needle.width,
            ytop=needle.y+needle.height,
            ybottom=needle.y)

# creating and confirming a subset of haystack that matches needle
# produces a graph of images side-by-side

red.layer <- 1 # We're only interested in one layer
haystack.topImage <- haystack.height-needle.y-needle.height
haystack.brick <- haystack.raster[haystack.topImage:(needle.height+haystack.topImage),needle.x:(needle.width+needle.x),red.layer]

plot(c(0, haystack.width), c(0, haystack.height), main=date(),type = "n")

rasterImage(haystack.brick,
            xleft=0,
            xright=ncol(haystack.brick),
            ytop=nrow(haystack.brick),
            ybottom=0)

rasterImage(needle.brick,
            xleft=2000,
            xright=(needle.width+2000),
            ytop=nrow(haystack.brick),
            ybottom=0)

# This shows a 100% correlation between two identical images
ccf(as.vector(needle.brick),as.vector(needle.brick),main=date())

# This shows the difference between two similar images
cor1 <- ccf(as.vector(haystack.brick),as.vector(needle.brick),main=date())
max(cor1$acf) # 0.7273756

# This shows the difference between two dissimilar images
ccf(as.vector(haystack.raster),as.vector(needle.brick),main=date())
```

III. CONCLUSION

In this technologically advanced world, it has become necessary for the beginners in R and image analysis to learn and understand the things very clearly, so that they could be applied and used in the future. Hence, we have performed the Image analysis tasks in R programming language, so that it becomes easy to all to understand the concepts related to it. This paper also provides the use of R Image Library (RIL), using which we can prominently develop the R based image processing software and can be useful for number of applications like remote sensing, agriculture, space center, satellites, medical and health sciences, etc. Thus, it can be concluded that R and Image analysis proves to be the better combination for learning, developing and understanding the capabilities provided in it.

REFERENCES

[1] Harshada, Ms & Snehal, & Shitole, Sanjay & Pranaya, Mhatre & Suchita, Kadam & Shweta, Ghanate & Kurle, Darshana. (2016). Python Based Image Processing.

[2] Mendoza, Fernando & Lu, Renfu. (2015). Basics of Image Analysis. 10.1007/978-1-4939-2836-1_2.

[3] Poningsih, Poningsih. (2017). Design of the expert system to analyze disease in Plant Teak using Forward Chaining. International Journal of Artificial Intelligence Research. 1. 11. 10.29099/ijair.v1i1.11.

[4] Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. (2015) "Going deeper with convolutions." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1-9. DOI: 10.1109/CVPR.2015.7298594

[5] Su, Hang, Shaogang Gong, Xiatian Zhu, Adrian Popescu, Alexandru Ginsca, Herve Le Borgne, Yuen Peng Loh et al. (2017) "WebLogo-2M: Scalable Logo Detection by Deep Learning From the Web." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 270-279. DOI: 10.1109/CVPR.2015.7298670.

[6] Forder L, Taylor O, Mankin H, Scott RB, Franklin A. (2016) Colour Terms Affect Detection of Colour and ColourAssociated Objects Suppressed from Visual Awareness. Ben Hamed S, ed. PLoS ONE.;11(3):e0152212. doi:10.1371/journal.pone.0152212..

[7] Joulan K, Brémond R, Hautière N. (2015) Towards an Analytical Age-Dependent Model of Contrast Sensitivity Functions for an Ageing Society. The Scientific World Journal:625034. doi:10.1155/2015/625034.

[8] Nasrollahi, Kamal, Sergio Escalera, Pejman Rasti, Gholamreza Anbarjafari, Xavier Baro, Hugo Jair Escalante, and Thomas B. Moeslund. (2015) "Deep learning based super-resolution for improved action recognition." In Image Processing Theory, Tools and Applications (IPTA), 2015 International Conference on, pp. 67- 72. IEEE. DOI: 10.1109/IPTA.2015.7367098

[9] Savioja, Lauri, Akio Ando, Ramani Duraiswami, Emanuel AP Habets, and Sascha Spors. (2015) "Introduction to the issue on spatial audio." IEEE Journal of Selected Topics in Signal Processing 9, no. 5: 767-769. DOI: 10.1109/JSTSP.2015.2447112

LINK VIDEO PRESENTASI:
https://youtu.be/rZwOXU359yI